

# ANALIZA SISTEMA

## 1. Klase:

Korisnik, Administrator, Vlasnik, Gost, Clan, Vozilo, IGoogleMaps, IPaypal, ICreditCard, ParkingLokacija, ParkingDAO, EParking, Zahtjev, NacinPlacanja

### I. Korisnik(apstraktna klasa):

- Atributi:

imePrezime(string), username(string), password(string), JMBG(string), adresa(string), brojMobitela(string), email(string), vozilo(Vozilo)

- Operacije:

Konstruktor(prima sve attribute), getter, setter

### II. Clan:

- Atributi:

Naslijedjeni iz klase Korisnik, statusClanarine(Enum - ACTIVE/INACTIVE), rezervisanoParkingMjesto(null ako nije placena clanarina tj. Ako ima neaktivan status)(integer)

- Operacije:

Konstruktor(prima sve attribute), getter, setter

### III. Gost:

- Atributi:

Naslijedjeni iz klase Korisnik (nema dodatnih atributa)

- Operacije:

Konstruktor

### IV. Administrator:

- Atributi:

username(string), password(string)

- Operacije:

Konstruktor(prima sve parametre), getter, setter, registrujParking(ParkingLokacija), registrujVlasnika(Vlasnik), obrisiVlasnika(Vlasnik), obrisiParking(ParkingLokacija), obrisiClana(Clan)

### V. Vlasnik:

- Atributi:

username(string), password(string), imePrezime(string), prihodi(double), zahtjevi(List<zahtjev>)

- Operacije:

Konstruktor(prima sve parametre), getter, setter, obradiZahtjev(Zahtjev)

#### **VI. Vozilo:**

- Atributi:

modelAuta(string), brojTablice(string), brojSasije(string), brojMotora(string)

- Operacije:

Konstruktor(prima sve parametre), getter, setter

#### **VII. ParkingLokacija:**

- Atributi:

naziv(string), adresa(string), cijena(double), vlasnik(Vlasnik), kapacitet(integer), brojSlobodnihMjesta(integer)

- Operacije:

Konstruktor(prima sve parametre), getter, setter

#### **VIII. Zahtjev:**

- Atributi:

korisnik(Korisnik), vozilo(Vozilo)

- Operacije:

Konstruktor(prima sve parametre), getter, setter

#### **IX. EParking:**

- Atributi:

gosti(List<Gost>), clanovi(List<Clan>), vlasnici(List<Vlasnik>), administrator(Administrator), parkinzi(List<ParkingLokacija>)

- Operacije:

Konstruktor bez parametara, getteri, setter, izracunajRutu(), registrujKorisnika(Korisnik), izracunajCijenuParkinga(ParkingLokacija, Korisnik), logInKorisnika(Korisnik), logInVlasnika(Vlasnik), logInAdministratora(Administrator), registrujParking(ParkingLokacija), registrujVlasnika(Vlasnik), obrisiVlasnika(Vlasnik), obrisiParking(ParkingLokacija), obrisiClana(Clan), izracunajZaradu(ParkingLokacija), registrujUlazak(ParkingLokacija, Korisnik), registrujIzlazak(ParkingLokacija, Korisnik), prikaziInfoParkinga(ParkingLokacija), validacijaClanskogStatusa(Clan), platiParking(Korisnik, NacinPlacanja)

#### **X. NacinPlacanja:**

Enum: PAYPAL, CREDITCARD, CASH

## XI. StatusClanarine:

Enum: ACTIVE, INACTIVE

## XII. ParkingDAO:

Baza podataka(sadrzi korisnicke naloge, parking lokacije)

## 2. Veze izmedju klasa:

- Vezu tipa generalizacije mozemo uociti izmedju klase Korisnik i klasa Clan i Gost, jer su Clan i Gost naslijedjeni iz klase Korisnik.
- Vezu tipa agregacije mozemo uociti izmedju klasa ParkingLokacija i Vlasnik, jer ParkingLokaciji mozemo dodijeliti drugog vlasnika, takodjer izmedju klasa Vlasnik i Zahtjev jer zahtjev moze postojati bez Vlasnika.
- Vezu tipa kompozicije mozemo uociti izmedju klasa EParking i Korisnik, Administrator i Vlasnik, jer Korisnik, Administrator i Vlasnik ne mogu postojati bez EParkinga, takodjer izmedju klasa Vozilo i Korisnik jer Vozilo ne moze postojati bez Korisnika.
- Veze tipa realizacije mozemo uociti izmedju klasa EParking i Sistema IGoogleMaps, IPayPal i ICreditCard jer klasa EParking implementira metode navedenih interface-a.

## 3. SOLID principi:

- **Single responsibility Principle** – svaka klasa ima samo jednu ulogu, jedina koja bi mogla narusiti ovaj princip je klasa EParking (ona vrši npr. plaćanje).
- **Open/Closed Principle** – ovaj princip je narušen jer klasa EParking ima kao atribut druge klase (Korisnik, Administrator, Vlasnik, ParkingLokacija). ParkingLokacija ima klasu Vlasnik u sebi i klasa Vlasnik ima klasu Zahtjev, a klasa Korisnik ima klasu Vozilo.
- **Liskov Substitution Principle** – u našem slučaju je zadovoljen jer na svim mjestima gdje koristimo baznu klasu Korisnik mozemo koristiti izvedene klase Gost i Clan.
- **Interface Segregation Principle** – ovaj princip je zadovoljen jer interdice-i IGoogleMaps, IPayPal i ICreditCard obavljaju samo jednu vrstu akcija, IGoogleMaps obavlja funkciju racunanja rute, IPayPal obavlja funkciju placanja pomocu Paypal racuna, a ICreditCard obavlja funkciju placanja pomocu kartice.
- **Dependency Inversion Principle** – ovaj princip je ispostovan jer bazna klasa Korisnik je ujedno i apstraktna klasa.